Managing the Test Process

A. Hamam

Overview

- Many Organization leave testing to the end
  - Testing is compromised
  - Not enough time and budget remaining
  - Testing is done in an ad hoc manner
- Cannot add quality to a bad product in the last minute
  - Quality must accompany the process from the beginning
- Testing activities can be carried out with development stages
  - Within the confines of any specific development process
- Goal of activities is to detect or prevent faults throughout the software development process

Who does the testing

- Developers vs. an independent team

Testing Activities

- Should be incorporated into the beginning of software development
  - Leaving them till the end produces unreliable software
- Proceed through all subsequent s/w development steps
- Level 4 organizations
- This way dramatic improvement on the effectiveness and efficiency of testing.

SDLC

- Waterfall
- V-model
- Spiral
- Prototyping
- XP
- Scrum
- ……etc.

Common stages

- Requirements analysis and specification
- System and software design
- Intermediate design
- Detailed design
- Implementation
- Integration
- System deployment
- Operation and maintenance

Mistakes

- Can happen because any software process
  - Involves communication, comprehension, and transition of information
- Can happen at any time
  - Information handling
  - Transfer of information from one stage to another
  - Integrating components
- Testing should happen on a parallel track that affects all stages
- Testing should be embedded within every aspect of software development
- Testers should be embedded within all teams.

Sub-objective of testing

- Test actions
  - Testing the product or artifact created at that stage
- Test design
  - Using the development artifacts of that stage or testing artifacts from a previous stage to prepare to test the final software
- Test influence
  - Using development or test artifacts to influence future development stages

Requirements analysis and specification

- Test Action Goal
  - Evaluate the requirements
    - Correct, testable and complete
- Test Design Goal
  - Prepare for system testing and verification
  - Develop test criteria from requirements
  - High level test plans that outline testing strategy
  - Scope and objective for every stage
  - Support s/w needed for testing at every stage
- Test Influence Goal
  - Influence SAD
  - Test scenarios and plans build to show the system meet requirements
    - Detect conflicting and ambiguous requirements
    - Help architects develop a design that is easily testable.

Table 6.1. Testing objectives and activities during requirements analysis and specification

Objectives	Activities
Ensure requirements are testable	Set up testing requirements
Ensure requirements are correct	■ testing criteria
Ensure requirements are complete	■ support software needed
Influence the software architecture	■ testing plans at each level
	■ build test prototypes
	Clarify requirement items and test criteria
	Develop project test plan

System and software design

Table 6.2. Testing objectives and activities during system and software design

Objectives	Activities
Verify mapping between requirements specification and system design	Validate design and interface
	Design system tests
Ensure traceability and testability	Develop coverage criteria
Influence interface design	Design acceptance test plan
	Design usability test (if necessary)

- Test Action Goal
  - Mapping between requirements specification and design
  - Changes to req. spec should be reflected in corresponding design change
  - Tests help validate the design and interface
- Test Design Goal
  - Prepare for acceptance and usability testing
  - Acceptance test plan is created that includes
    - Acceptance test requirement
    - Test Criteria
    - Testing method
    - Spec and design should be kept traceable and testable
    - Prepare for unit testing and integration testing by choosing coverage criteria
- Test Influence Goal
  - Influence the design of the UI
  - Usability tests or an interface prototype clarify what the customer wants
  - Esp. important when UI is an integral part

Intermediate Design

Table 6.3. Testing objectives and activities during intermediate design

Objectives	Activities
Avoid mismatches of interfaces	Specify system test cases
Prepare for unit testing	Develop integration and unit test plans
	Build or collect test support tools
	Suggest ordering of class integration

- Test Action Goal
  - Problem in large software systems: Component interface mismatch
  - Avoid mismatches of interfaces.
- Test Design Goal
  - Prepare of unit testing, integration testing and system testing
  - Unit and integrations test plans are refined with information about interfaces and design decisions
  - Test support tools such as drivers, stubs and measurement tools should be acquired or build
- Test Influence Goal
  - Influence detailed design.
  - What order will components be eventually integrated and tested.

Detailed Design

Table 6.4. Testing objectives and activities during detailed design

Objectives	Activities
Be ready to test when modules are ready	Create test cases (if unit) Build test specifications (if integration)

- Test Action Goal
  - All materials are ready for testing when the modules are written
  - Prepare for unit and integration testing
  - Refine detailed test plans
  - Generate test cases for unit testing
  - Write detailed test specification for integration testing
- Test Influence Goal
  - Influence the implementation and unit and integration testing

Implementation

Table 6.5. Testing objectives and activities during implementation

Objectives	Activities
Efficient unit testing	Create test case values
Automatic test data generation	Conduct unit testing
	Report problems properly

- Test Action Goal
  - Effective and efficient unit testing
  - Effectiveness based largely on coverage criteria and test data generated
  - Unit testing performed according to what was decided earlier
    - unit testing plan
    - test criteria
    - Test cases
    - Test support tools
  - Unit test results and problems should be saved and reported properly
  - Designers and developers could be made available to help testers
    - They will be done with the bulk of coding and designing.
- Test Design Goal
  - Prepare for integration and system testing
- Test Influence Goal
  - Efficient unit testing can help ensure early integration and system testing
    - (cheaper to fix bugs at unit testing rather than system testing)

Integration

Table 6.6. Testing objectives and activities during integration

Objectives	Activities
Efficient integration testing	Perform integration testing

- Test Action Goal
  - Perform integration testing
  - Can begin once the unit testing passed
  - Decide ahead of time on order of classes to be integrated (more efficient)
    - Finish in that order
  - Another less efficient way is to integrate classes as soon
- Integration testing is concerned with finding errors that result form unexpected interactions among components

System Deployment

Table 6.7. Testing objectives and activities during system deployment

Objectives	Activities
Efficient system testing	Perform system testing
Efficient acceptance testing	Perform acceptance testing
Efficient usability testing	Perform usability testing

- Test Action Goal
  - System testing, acceptance testing and usability testing
  - System testing: compare the software system to its original objective
    - Validating the software meets the functional and non-functional requirements
  - System test cases are developed from the system and project test plan
    - From the requirements spec. and software design phase according to criteria
  - Acceptance testing can be started as soon as system testing is completed.
  - Acceptance testing ensures that the completed system satisfied the customers needs and should be done with their involvement
  - Test cases are derived from acceptance test plans and test data set up previously.
  - Usability testing evaluates the suer interface of the software
  - It should also be done with user involvement.
    - Usability testing to be covered in another s/w eng. Course.

Operation and Maintenance

Table 6.8. Testing objectives and activities during operation and maintenance

Objectives	Activities
Efficient regression testing	Capture user problems Perform regression testing

- Regression testing to make sure functionality is not broken (more on that in a later topic)

Final Remarks

- Put quality first
- No shortcuts even with time-driven management
- Test artifacts should be managed
  - Version control
  - Easily available
  - Updated regularly
- Artifacts include
  - Test design
  - Tests
  - Test results
  - Automated support
- Track source
  - When it changes the tests need to change